

# Optimal and Power Aware BIST for Delay Testing of System-On-Chip

Deepa Jose<sup>1</sup>, Dr. Nirmal Kumar P.<sup>2</sup>, Jayakumar P<sup>3</sup>

<sup>1,2,3</sup> Department of Electronics and Communication Engineering, College of Engineering Guindy,  
Anna University, Chennai - 600 025, India

1Email: deepajose11@gmail.com

**Abstract**—Test engineering for fault tolerant VLSI systems is encumbered with optimization requisites for hardware overhead, test power and test time. The high level quality of these complex high-speed VLSI circuits can be assured only through delay testing, which involves checking for accurate temporal behavior. In the present paper, a data-path based built-in test pattern generator (TPG) that generates iterative pseudo-exhaustive two-patterns (IPET) for parallel delay testing of modules with different input cone capacities is implemented. Further, in the present study a CMOS implementation of low power architecture (LPA) for scan based built-in self test (BIST) for delay testing and combinational testing is carried out. This reduces test power dissipation in the circuit under test (CUT). Experimental results and comparisons with pre-existing methods prove the reduction in hardware overhead and test-time.

**Index Terms**—delay testing, parallel BIST, digital VLSI, FPGA, fault tolerance, low power.

## I. INTRODUCTION

Current system on chip (SoC) are high performance multi-core heterogeneous systems, which include reconfigurable resources such as field programmable gate arrays (FPGA) to increase design flexibility. These high speed SOC designs fabricated in the deep-submicron/ lesser sub-micron domains are prone to process variations and large spatial variations, caused by intrinsic aspects such as line-edge roughness, random dopant and body thickness fluctuations. These variations lead to large spreads in circuit delay and power, making them more susceptible to delay faults of the order of picoseconds.

Delay faults affect the timing of the circuit at its operating speed. These faults can limit the continued scaling of silicon fabrication technologies. Design for testability (DFT) and fault tolerant design for delay testing needs to be addressed at the early stages of the VLSI design cycle. Path delay and transition delay are the most popular delay fault models, both of which requires two-pattern test for delay fault detection. The test requires a pair of vectors {T1, T2} to be applied at speed, where T1 initializes the target node/path and T2 launches the appropriate transition and propagates it to an observable point.

Iterative structures suited for VLSI implementations like multipliers, digital signal processing systems, bit-sliced microprocessors and embedded memories require IPET for delay testing. Iterative pseudo-exhaustive one-pattern generators are proposed in [1] and [2]. No scheme for

accumulator based IPET test generation exists except in [3]. In this paper, less complex circuits already existing in the SOC are utilized for efficient BIST applications. The implementation of a built-in test pattern generator (TPG) for generating iterative pseudo-exhaustive two-patterns (IPET) is discussed. This TPG enables testing of iterative logic arrays as well as VLSI circuits partitioned for pseudo-exhaustive testing. The design aims at the reduction of test time, hardware overhead incurred due to DFT and optimal fault coverage. Very few dynamic test power reduction methods for scan based delay testing are discussed in the present available literature [4]. In [5], a method for minimizing peak power consumption during scan testing using x-filling heuristics is explained. In [6], a method for reducing the test power is scan BIST using dynamic clock control circuit is presented. Hence, in the present paper a CMOS implementation of a low power architecture (LPA) for scan based delay testing as well as combinational fault testing of VLSI systems is implemented. Further, the FPGA implementations of these architectures is carried out.

The structure of the paper is as follows: In section II, the preliminaries for pseudo-exhaustive testing and delay testing is given. Section III, IV, and V outlines the architectures and the analytical basis. Section VI presents the discussion of experimental results. Finally, Section VII summarizes the methodology and scope for future work.

## II. BACKGROUND

Many previous works prove that, external testing techniques using automatic test equipment (ATE) and pseudo-random testing, practiced by the industry, are inefficient for at-speed delay testing. An alternate to pseudorandom testing is exhaustive testing, which requires  $2^{2n}$  test patterns for delay testing, where 'n' is the number of inputs of the VLSI circuit. Apparently, true exhaustive testing is impractical for large VLSI circuits. Partitioning to allow exhaustive test of sub-circuits offers an attractive alternative, which leads to pseudo-exhaustive testing approach [1].

Pseudo-exhaustive testing provides 100% fault-coverage for detectable faults. For an 'n' input CUT with 'm' outputs and cone-size 'k', pseudo-exhaustive testing involves in applying exhaustive test to the 'm' output cones. In many cases, each output of (n, m, k) CUT depends only on a subset of primary inputs. A segment with a single output is called a cone. Here, the dependency set 'k' of an output represents the number of inputs on which an output depends. This CUT,

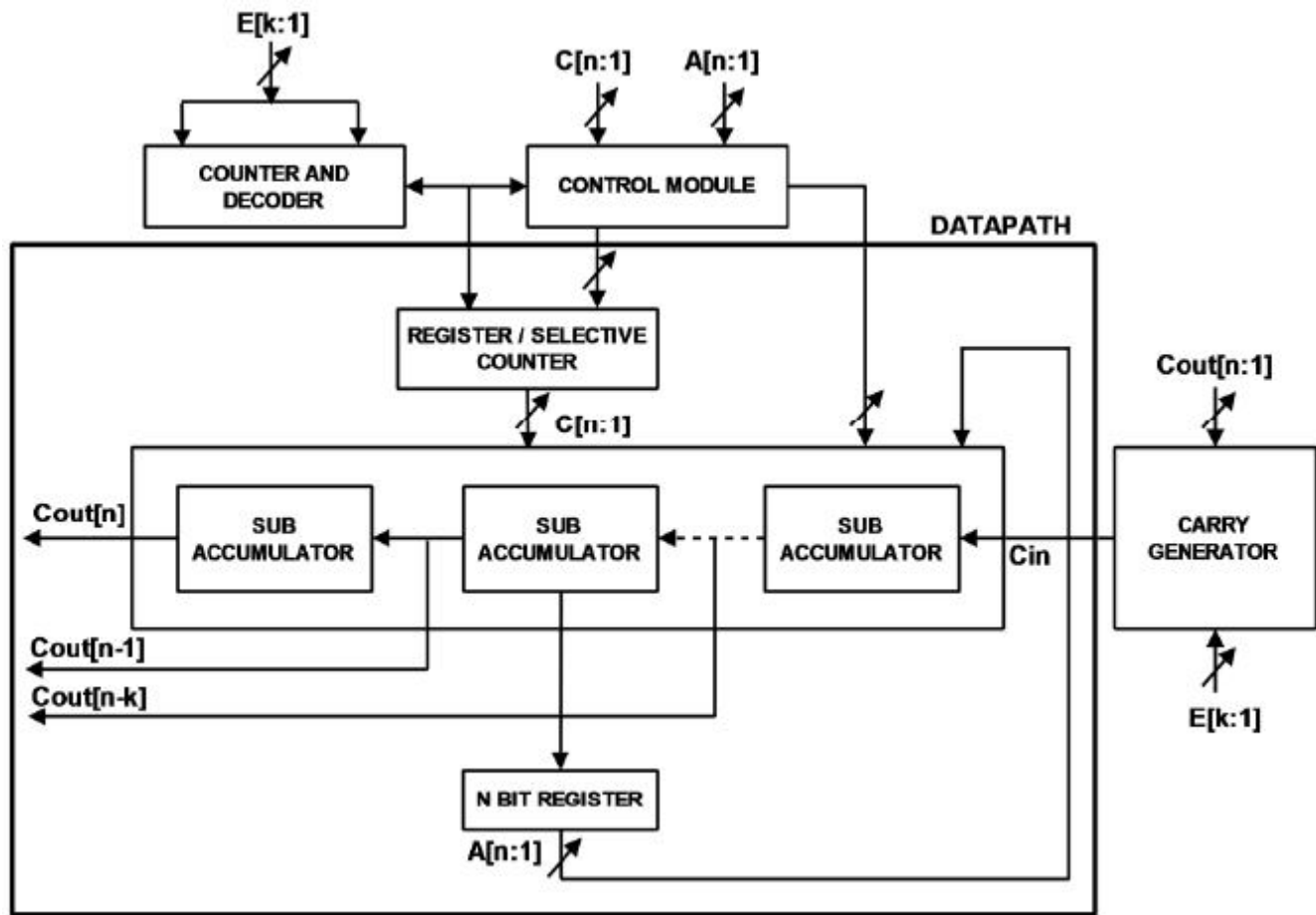


Figure 1. Architecture of the TPG

partitioned into 'm' cones with cone-size 'k', can be tested in parallel, thus reducing the total test time and test vector length. Parallel testing can be achieved by inserting suitable DFT logic at the partitioned points [7].

In CMOS VLSI circuits, dynamic power significantly contributes to total power dissipation. Dynamic power results from the activity of a circuit in changing its states due to the charging and discharging of the effective capacitive loads. Dynamic Power 'P' dissipated at a node is given by:

$$P = 1/2cv^2af \quad (1)$$

where 'c' is the capacitance of the node, 'v' is supply voltage, 'f' is the clock frequency and '±' is the node activity factor. Circuits are often designed to operate in normal and test modes. Dissipation of power is more in a test mode than a normal mode, especially if a scan mechanism is employed. This is due to random combinational logic activity in the CUT due to scan in and scan out of test data that subsequently increases the power consumption and test time. Yield loss can result if this consumption is higher than that of the normal functional operation for which the circuit is designed.

### III. THE TPG ARCHITECTURE

In IPET, the entire 'n' bit space will be exhaustively covered, if for all 'n-k+1' adjacent 'k' bit subspaces, all possible two-patterns occur at least once. The TPG shown in Fig. 1 is

a module with n-inputs  $E[n:1]$ , which are used to select cone-size 'k' of the IPET generated at the outputs of the accumulator. The TPG consists of two parts: data-path module and control module. The control module controls the entire IPET generation process. The selective counter is designed so that, its increment value in each clock cycle for every signal  $E[k]$  is as shown in Table I. The n-stage accumulator accumulates the output of selective counter. Depending on the values of  $E[k]$ , the n-stage selective counter and accumulator are reconfigured to work as k-sub-stage counters and k-sub-stage accumulators respectively. The carry input of each sub-accumulator is driven by the carry output of the preceding sub-accumulator. The  $E[k]$  signals and the carry outputs of the n-stage generic accumulator form the input of carry generator module. In this module, when  $E[k]$  is enabled, the corresponding  $Cout[k]$  signal of the n-stage accumulator is propagated as carry input ( $Cin$ ) to the accumulator. As depicted in Fig. 2, the control module detects specific states of the selective counter and accumulator, to generate the required control signals for the entire architecture.

After each (n,k) pseudo-exhaustive two-pattern test is complete, 'endtest' signal is generated and 'k' is incremented until  $k \leq n$  to iterate the two-patterns. For this purpose, the 'endtest' signal clocks a  $\lceil \log_2 n \rceil$  stage counter which drives the inputs of a  $\lceil \log_2 n \rceil : n$  decoder. General pseudo-exhaustive two-patterns are obtained by excluding these two blocks.

TABLE I. FUNCTIONALITY OF TPG GENERATOR

$E[k]=1$	Output of sub-stage accumulator(n)	In each clock selective counter(n) increased by
E[1]	(XXXXXXXXXXXX)	0000 0000 0001
E[2]	(X) (X) (X) (X) (X) (X) (X) (X) (X) (X) (X) (X)	1111 1111 1111
E[3]	(XX) (XX) (XX) (XX) (XX) (XX)	0101 0101 0101
E[4]	(XXX) (XXX) (XXX) (XXX)	0010 0100 1001
E[5]	(XXXX) (XXXX) (XXXX)	0001 0001 0001
E[6]	(XX) (XXXXXX) (XXXXXX)	0100 0010 0001
E[7]	(XXXXXXXX) (XXXXXXXX)	0000 0100 0001
E[8]	(XXXXXXXX) (XXXXXXXX)	0000 1000 0001

## IV. CONTROL MODULE OPERATION

The control module operation which forms the crux of the design is given below. Its operation is based on two-pattern test algorithm proposed in [8]. The algorithm generates all n-bit two-pattern test transitions within  $2n \times (2n-1) + 1$  cycles. The algorithm consists of three steps: Generation of n-bit s-circle(2n-3) starting from (2n-1)

- Generation of n-bit circle(2n-2) starting from (2n-1)
- Generation of zero-transitions

where,

- S-circle(i) - generates vectors starting from A and performing consecutive sequence(i) until returning to A after the final step(i) of sequence(i).
- Step(i) - generates transitions from number A to number B such that  $B = (A+i) \bmod N$
- Circle(i) - generates vectors originating from A and performing consecutive step(i) until returning to number A.
- Sequence(k) - generates vectors starting from A and performing consecutive step(i) such that,  $i = 1, 2, 3, \dots, k$ .

Assuming  $n=4, k=3, N=2n$  and  $K=2k$ , the control module operation is as per the following phases (refer Fig. 2):

- Phase 1: In this phase, selective counter starts counting in each clock cycle and accumulator accumulates the count value. This continues until the selective counter reaches  $K-3=5$ . When selective counter value is 5 (fifth clock cycle), it is reset and count is resumed in the sixth clock cycle. For this purpose, the control logic should generate 'resetc' control signal. Accumulator continues the accumulation process and enters phase 2.
- Phase 2: In this phase, when accumulator value is  $N-1=15$  and selective counter value is  $K-3=5$  (35th clock cycle), the selective counter value is incremented by one clock cycle and is disabled. The algorithm enters phase 3. The control module has to generate 'hold' signal to disable the counter.
- Phase 3: In this phase, when accumulator again holds the value  $N-1=15$  (42nd clock cycle), clock of the selective counter is divided by 2. The accumulator continues to accumulate and is reset to zero every second clock cycle. During this phase, the control module has to generate accumulator reset

signal 'reseta' and a 'dividebytwo' signal for the selective counter. This process continues until selective counter and the accumulator reaches  $N-1=7$  (56th clock cycle). For this condition, the control module has to generate endtest signal, to indicate end of (n,k)-pseudo-exhaustive two-pattern test.

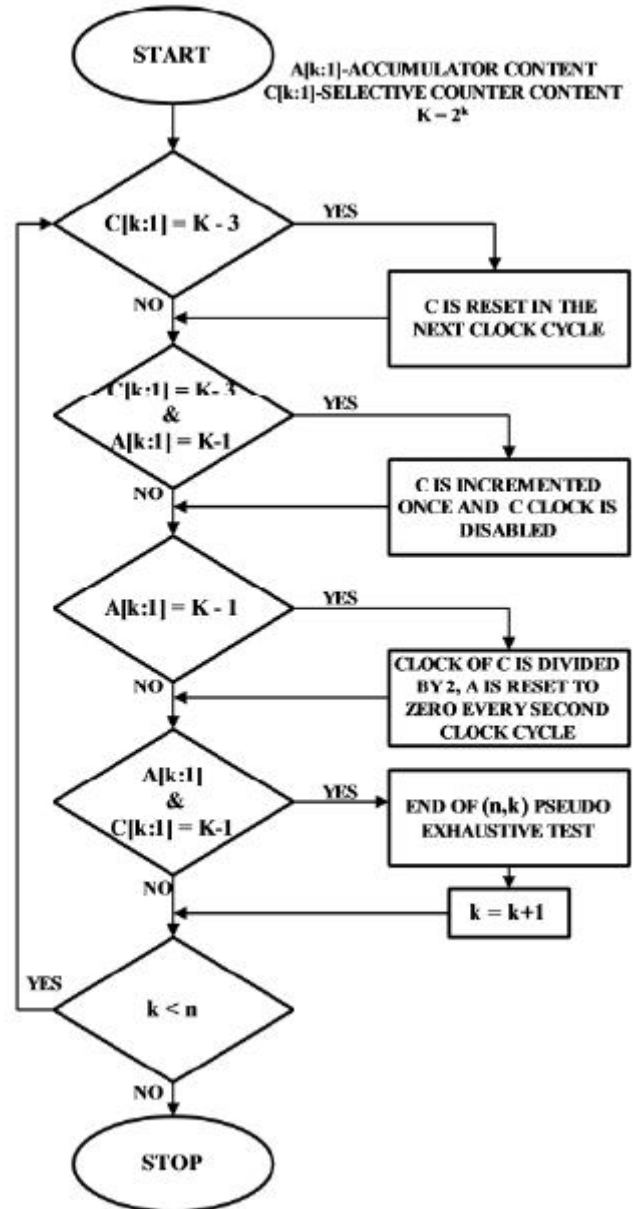


Figure 2. Flowchart of control module operation

## V. LOW POWER ARCHITECTURE FOR DELAY TESTING

In the present work, the low power architecture (LPA) reduces the number of switching transitions from propagating into the CUT, subsequently reducing the dynamic power. This is achieved by modifying the scan chain using a separate C-chain of flip-flops and a logic array of XOR gates and AND gates as shown in Fig. 3 [9]. Test data is transferred from the C-chain to the D-register cells. Cells in the C-chain change only in the shift mode. In this mode, the enable signal (clke) is 0. Therefore, transitions in the C-chain are masked by the enable signal (clke) of the AND gate. The transitions can

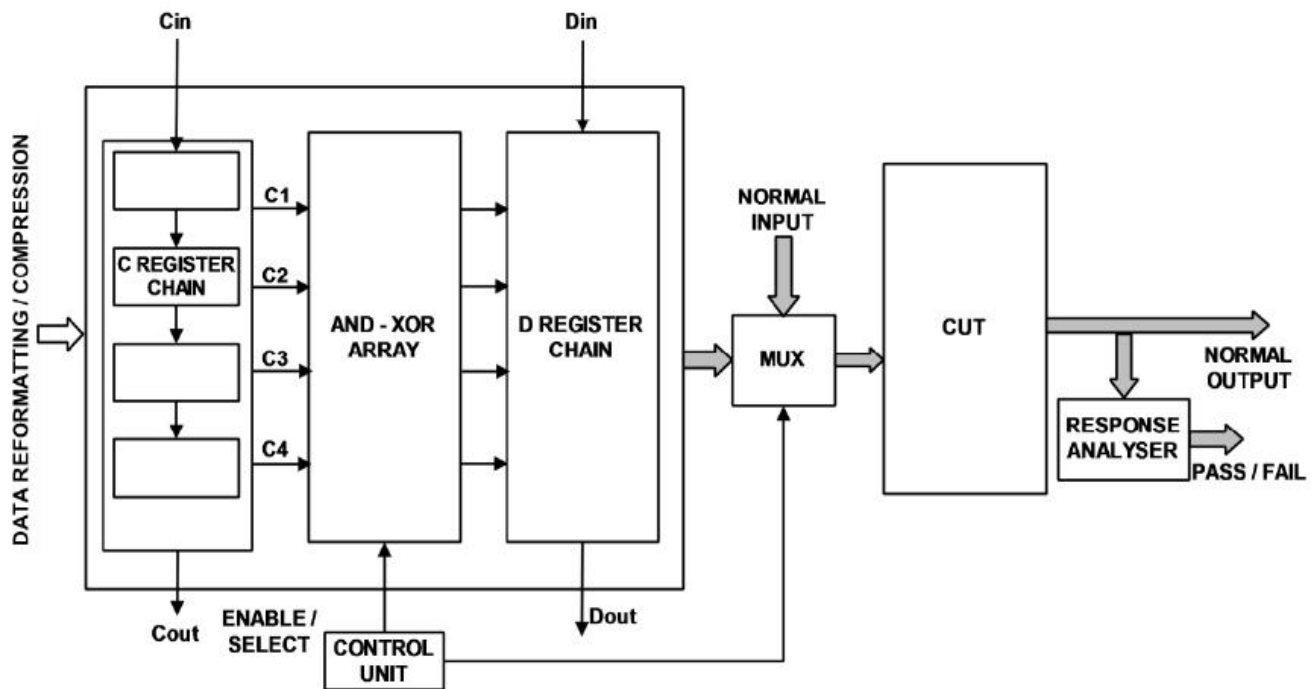


Figure 3. The low power architecture

only affect the XOR gate. This causes a small number of transitions at the data register outputs and hence low test power dissipation in the CUT. As most flip-flops provide both Q and its complement Q0, XOR functionality can be provided by using two transistors in a pass transistor-based structure (refer Fig. 11). The effects of this implementation on power consumption are negligible due to pass-transistor based structure. As shown in Fig. 11, the logic required for providing the enable signal (clke) signal requires only six transistors.

Testing a delay fault requires the application of different vectors at two consecutive clocks. Unlike the conventional scan cells, the LPA architecture allows the at-speed application of arbitrary vector pairs to the CUT in two consecutive clocks, resulting in testing of delay faults. This is accomplished as follows: Assume that the test generated for a delay fault includes the vector pair (V1,V2). First, the DR chain is loaded with V1. Then, the difference vector TD that is required for changing the D vector V1 to V2 is shifted into the C- chain. At the next clock edge, the D vector V1 is changed to V2. Furthermore, C-chain hardware is designed to take advantage of similar adjacent test data. Reconstruction of the original test vectors from the difference information is possible using the LPA. Consequently, this attribute makes the architecture suitable for algorithms that are based on compressing test vectors. LPA can reduce test time and power in conjunction with test reformatting techniques. Test data compression can additionally decrease test time and memory requirements.

## VI. SIMULATION ENVIRONMENT AND RESULTS

The discussed architectures are synthesized using Cadence encounter tool and technology mapping is done based

on 0.18 micrometer technology library. Comparisons in test length for exhaustive two-pattern generator and pseudo-exhaustive two-pattern generator for various values of 'n' and 'k' for optimal fault coverage are given in Table II.

Reduction in test-vector length, results in reduced test-time and test-power. Test-length depends on cone-size, regardless of the VLSI network size. From the generated patterns depicted in Table. III, it can be inferred that, identical test patterns are generated with different cone size depending on the value of E[k]. This leads to variations in the degree of parallelism for testing. Hence the TPG can be used to test systems with varying cone-sizes, thus easily varying the degree of parallelism for testing.

TABLE II. TEST-LENGTH COMPARISONS

n	Exhaustive Two- Pattern Test Length	Pseudo Exhaustive Two- Pattern n, k      Test Length		% reduction in length of test vectors
4	256	4,2	13	94.92
5	1024	5,3	57	94.43
6	4096	6,2	13	99.68
7	16384	7,3	57	99.65
8	65536	8,2	13	99.98

The control module is designed by representing its functionality using a finite state machine (FSM). Design entry is done by a behavioral description of its functionality using verilog hardware description language (VHDL). Synthesis and optimization for different values of „n is carried out using Cadence Encounter tool. Carry generator module is

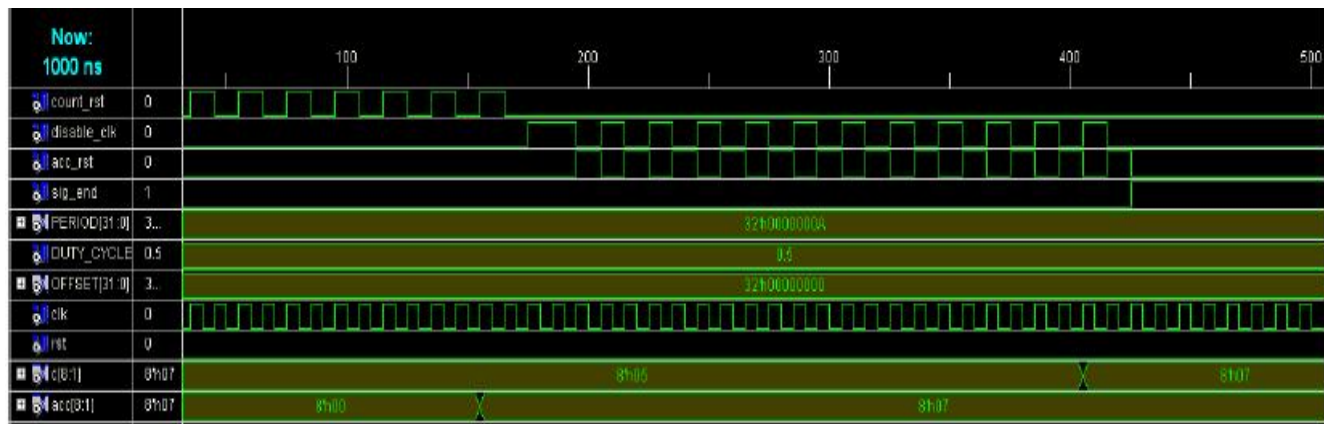


Figure 4. Generation of control signals

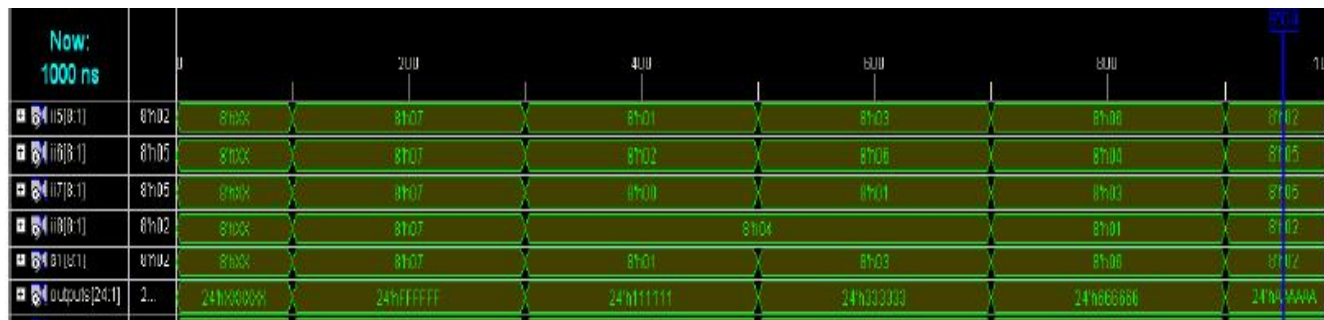


Figure 5. Generation of IPET patterns

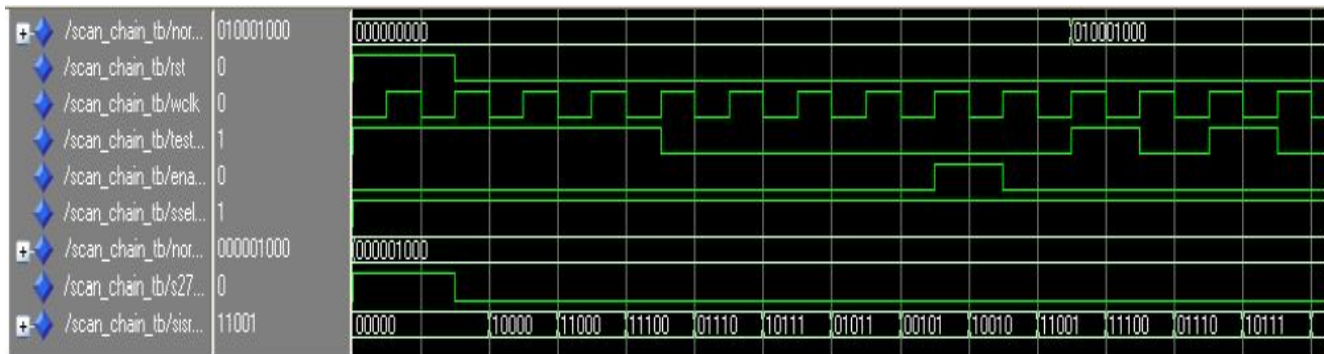


Figure 6. Signature obtained for s27 benchmark circuit with stuck-at-1 fault

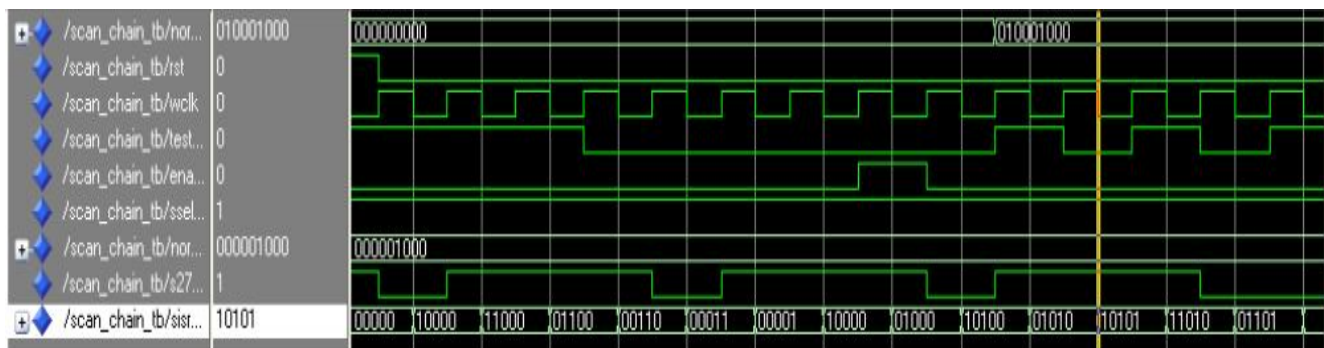


Figure 7. Signature obtained for fault free s27 benchmark circuit

designed with 'n' AND gates and an OR gate. Design of 'm to n' decoder and 'm' stage counter is done with '2n' gates with 'm' D flip-flops. Selective counter design is obtained by modifying D-registers in the data-path with 'n' 2:1 multiplexers

and '2n' OR gates [3]. The RTL schematics of the synthesized modules and simulation results are shown in Fig. 3-6.

Very few IPET generators are proposed for hardware overhead comparisons. Hence, one option is to multiplex the outputs of two pseudo-exhaustive one-pattern generators



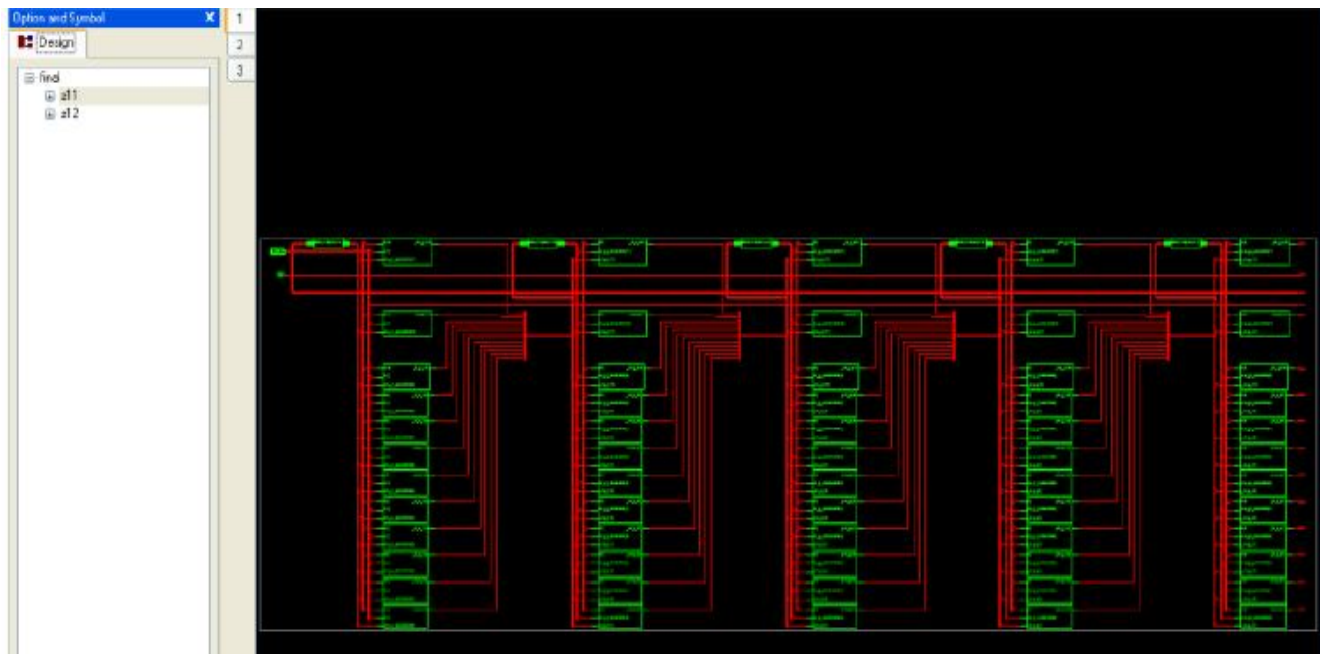


Figure.8 RTL schematic of selective counter

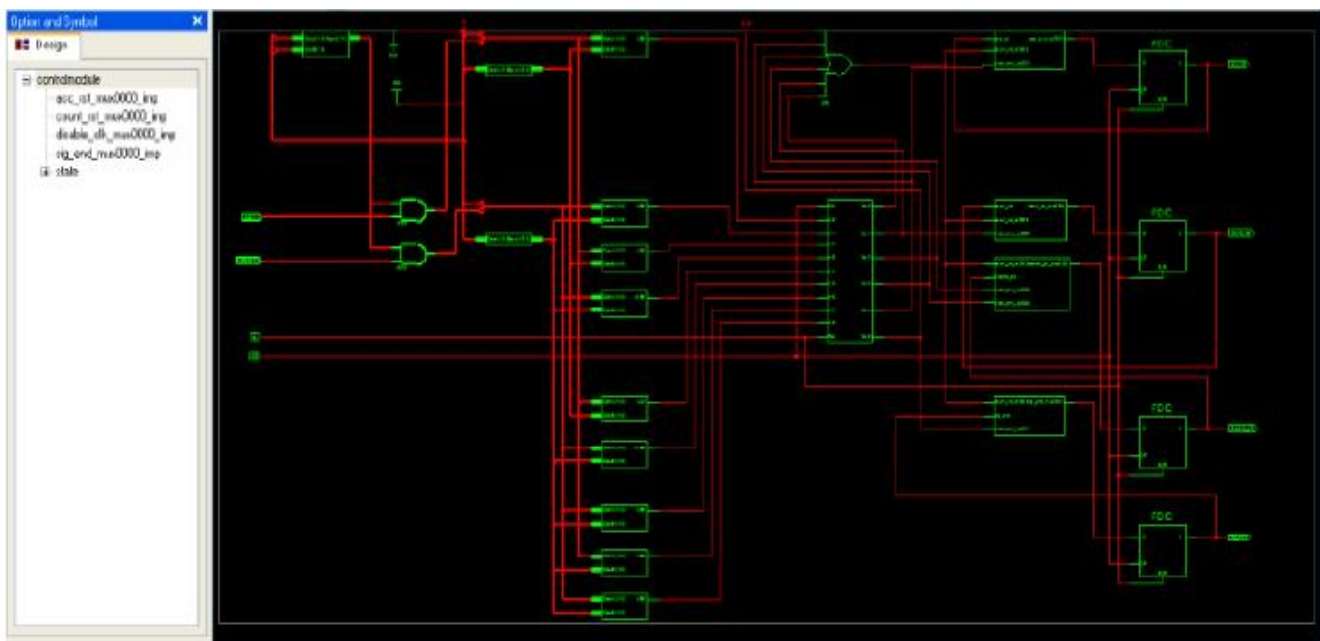


Figure 9. FPGA implementation of control module

proposed in [1] and [2]. Hardware overhead comparisons are made assuming that XOR gate consists of 4-gate equivalents, D-flip-flop consists of 8-gate equivalents and multiplexer consists of 3-gate equivalents.

From Table IV, it can be concluded that compared with pre-existing works, there is a reduction in hardware overhead for the proposed TPG design. Datapath based TPG is realized on target device Spartan 6E field programmable gate array (FPGA). The synthesis results shows very less device utilization. Hence, this TPG can aid the BIST of FPGA systems. The reprogrammability of FPGAs can be utilized to create the BIST logic. The FPGA can be configured as parallel 1-D iterative logic arrays (ILA). By repeated reconfiguration, compre-

hensive fault coverage can be obtained. The BIST logic ceases to exist when the circuit is reconfigured for its normal system function. Thus, testability can be achieved without any area overhead or performance degradation.

The proposed LPA does not increase the delay during normal operation during delay fault testing. The generated test vectors are applied to s27 benchmark circuit and the output of CUT is applied to a signature analyzer. From the generated signature, any stuck-at-faults in the CUT can be identified. From the simulation results shown in Fig. 6 and Fig. 7, the difference in signature for fault-free and fault injected circuit is observed. Fig. 12 and Fig. 13 show the TSPICE simulation waveforms obtained from the CMOS implementa-

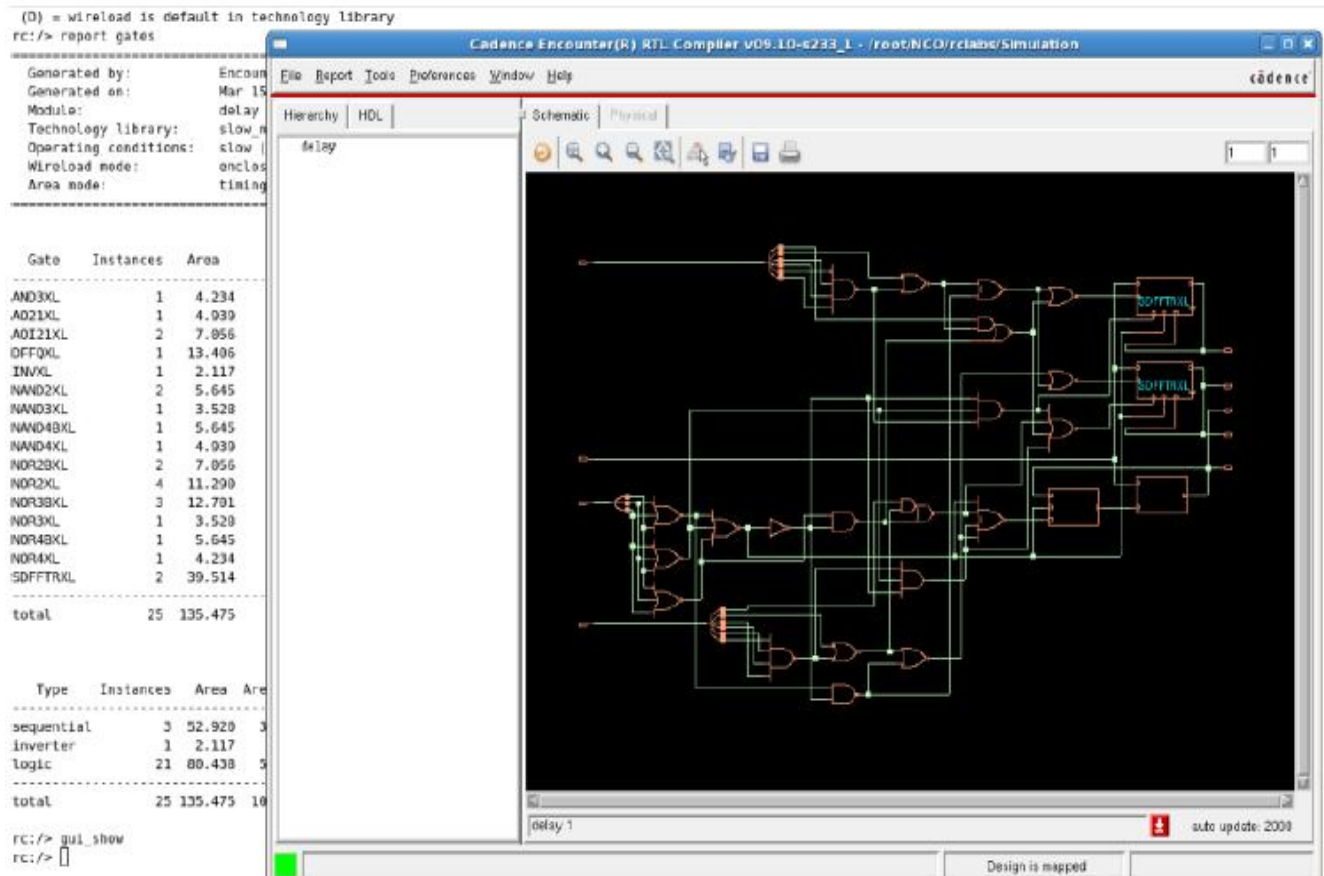


Figure. 10 Schematic obtained in Cadence for the control module

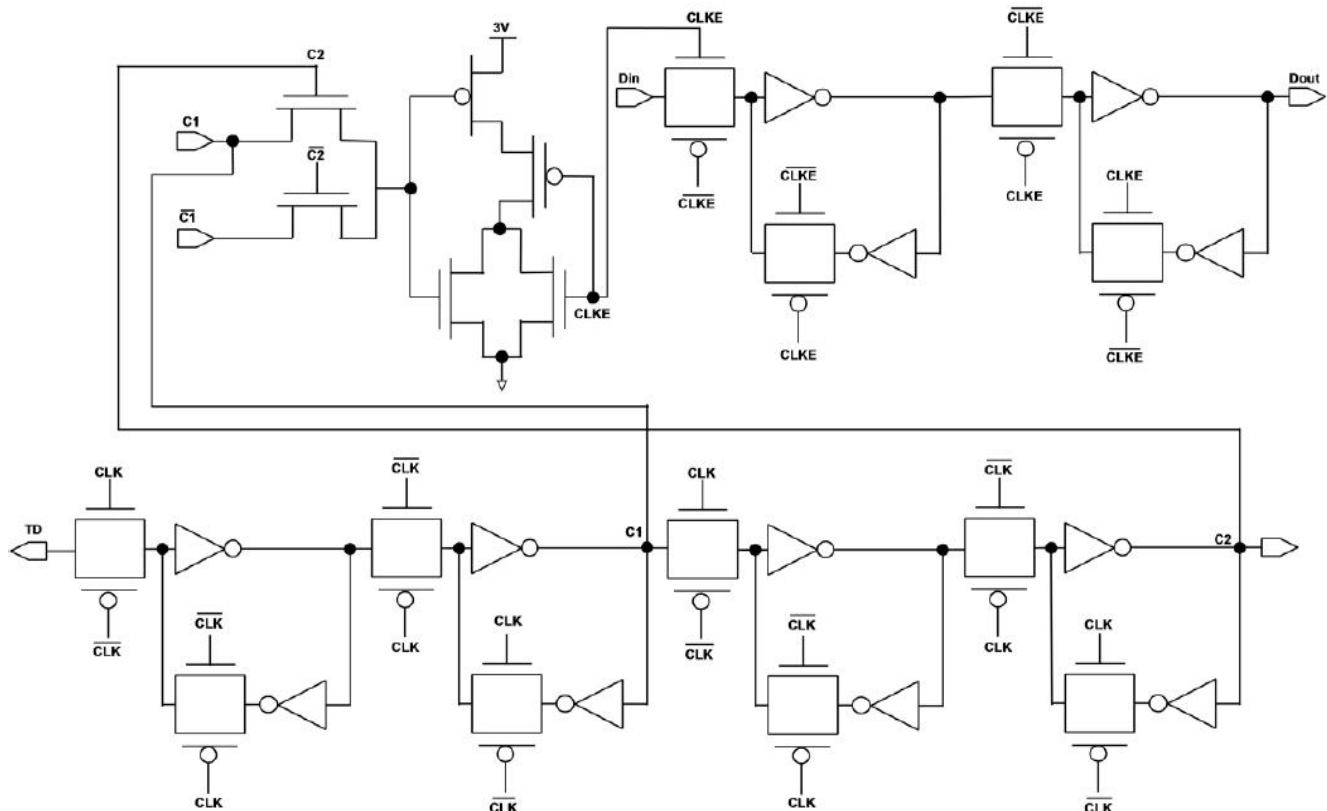


Figure 11. CMOS circuit level implementation of LPA

TABLE III. A PART OF IPET GENERATED BY TPG

OUTPUT(n=6, k=2)	Count	OUTPUT	Count	OUTPUT	Count
(11)(11)(11)	1	(010)(010)	11	(010)(010)	35
(01)(01)(01)	2	(011)(011)	12	(111)(111)	36
(10)(10)(10)	3	(101)(101)	13	(110)(110)	37
(11)(11)(11)	4	(001)(001)	14	(101)(101)	38
(10)(10)(10)	5	(101)(101)	15	(100)(100)	39
(01)(01)(01)	6	(001)(001)	16	(001)(001)	40
(11)(11)(11)	7	(100)(100)	17	(010)(010)	41
(00)(00)(00)	8	(110)(110)	18	(001)(001)	42
(01)(01)(01)	9	(010)(010)	19	(111)(111)	43
(00)(00)(00)	10	(110)(110)	20	(000)(000)	44
(10)(10)(10)	11	(100)(100)	21	(001)(001)	45
(00)(00)(00)	12	(101)(101)	22	(000)(000)	46
(11)(11)(11)	13	(111)(111)	23	(010)(010)	47
n=6, k=3		(011)(011)	24	(000)(000)	48
(111)(111)	1	(111)(111)	25	(001)(001)	49
(001)(001)	2	(101)(101)	26	(000)(000)	50

TABLE IV. HARDWARE OVERHEAD COMPARISONS

	[1]	[2]	[3]	[TPG]
N	gates	gates	gates	gates
5	194	138	109	90
6	204	164	126	111
7	214	190	143	120
8	224	208	160	154
11	390	374	219	218
13	410	442	253	249

tion of LPA with three scan cells. This simulation is for 0.25 micrometer CMOS technology library and a voltage supply of 3V. Dout (Vq1) is the output of the data register. TD (Cin1) is the serial input of the C-chain. C1(Cout1) and C2(Cout2) are the outputs of the two C-chain triggering registers. In this simulation, after shifting test data (Cin1) to the C-chain, the new test vector is triggered in the D register chain when the clke (clkenable) is active. The disadvantage of the LPA is the increased area-overhead due to the AND-XOR logic.

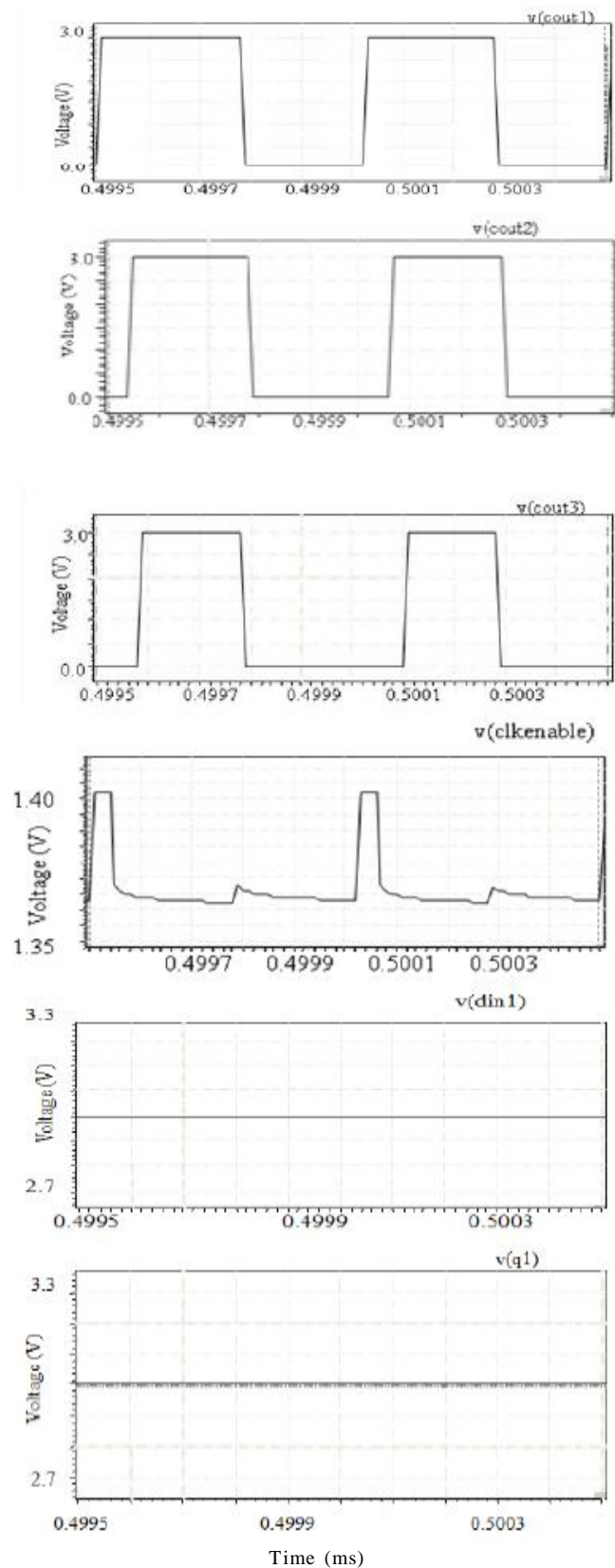
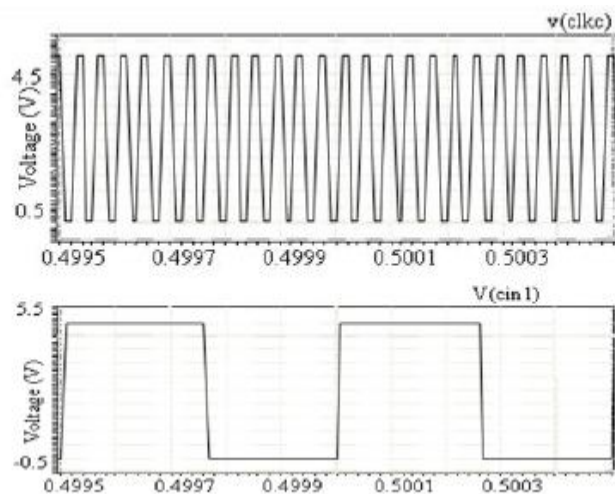


Figure 12. Simulation waveforms for LPA

## VII. CONCLUSION

In VLSI circuits, utilization of the same TPG for BIST of modules with different cone sizes drives down the DFT hard



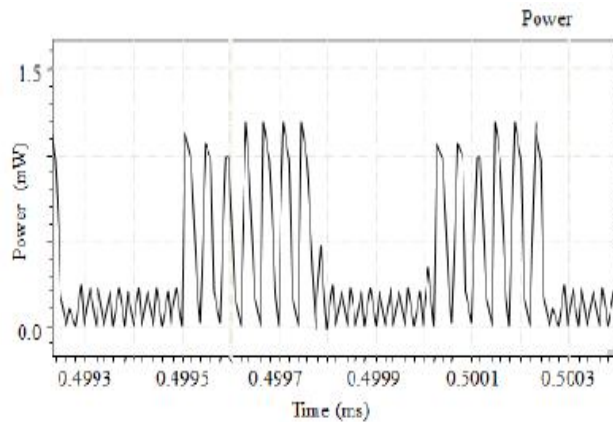


Figure13. Power results obtained for Low Power Architecture

ware overhead and DFT cost. The datapath based TPG presented in the current paper provides optimal fault coverage, at-speed parallel testing, reduction in hardware overhead and reduction in test time. In the future, this TPG will be used to aid the design of fault tolerant self-checking VLSI systems. The implementation of LPA for delay testing proves its improved performance in terms of test time and test power. In the future, the LPA performance will be improvised by applying efficient compression algorithms to the test data vectors.

#### REFERENCES

- [1] P. Dasgupta, S. Chattopadhyay, P. P. Chaudhuri, and I. Sengupta, "Cellular automata-based recursive pseudo-exhaustive test pattern generation," *IEEE Trans. Comput.*, vol. 50, no. 2, pp. 177–185, February 2001.
- [2] J. Rajski and J. Tyszer, "Recursive pseudoexhaustive test pattern generation," *IEEE Trans. Comput.*, vol. 42, no. 12, pp. 1517–1521, December 1993.
- [3] I. Voyiatzis, D. Gizopoulos and A. Paschalis, "Recursive Pseudo-Exhaustive Two-Pattern Generation," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 18, no. 1, January 2010.
- [4] P. Girard, "Survey of Low-Power Testing of VLSI Circuits," *IEEE Design and Test of Computers*, vol. 19, pp. 80–90, June 2002.
- [5] N. Badereddine, P. Girard, S. Pravossoudovitch, C. Landrault, and A. Virazel, "Minimizing Peak Power Consumption during Scan Testing: Test Pattern Modification with X Filling Heuristics," in *Proc. Int. Conf. on Design and Test of Integrated Systems in Nanoscale Technology*, pp. 359–364, September 2006.
- [6] P. Shanmugasundaram and V. D. Agrawal, "Dynamic Scan Clock Control in BIST Circuits," *Proc. 2nd IEEE Int. Conf. on Reliability Aware System Design and Test of Integrated Systems*, pp. 25-30, January 2011.
- [7] K. Kumar, S.U. Bhaskar, P.S. Chattopadhyay, P. Mandal, "Circuit Partitioning using Particle Swarm Optimization for Pseudo-Exhaustive Testing," *Proc. IEEE Int. Conf. Advances in Recent Technologies in Communication and Computing*, pp. 346-350, 2009.
- [8] I. Voyiatzis, A. Paschalis, D. Nikolos, and C. Halatsis, "Accumulator-Based BIST Approach for Two-pattern Testing," *Journal of Electronic Testing Theory and Applications*, vol. 15, no. 3, pp. 267-278, December 1999.
- [9] M. Hosseinabady, S. Sharifi, F. Lombardi, Z. Navabi, "A Selective Trigger Scan Architecture for VLSI Testing," *IEEE Transactions on Computers*, 57, 316–328, 2008.